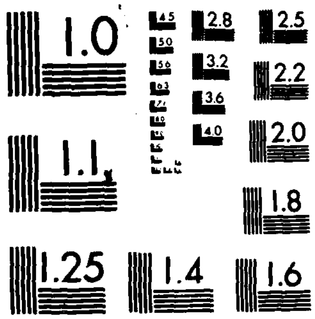
AD-A138 904 FINDIF: A SOFTWARE PACKAGE TO CREATE SYNTHETIC SEISMOGRAMS BY FINITE DIFFERENCES(U) WOODS HOLE OCEANOGRAPHIC INSTITUTION MA M M HUNT ET AL. NOV 83 1/1

UNCLASSIFIED WHOI-83-42 N00014-79-C-0071 F/G 8/11 NL

END
DATE
FILMED
4-84
DTIC


1.0
2.8
2.5
3.2
2.2
3.6
4.0
2.0
1.1
1.8
1.25
1.4
1.6

(12)

WHOI-83-42

Woods Hole Oceanographic Institution

WOODS HOLE OCEANOGRAPHIC INSTITUTION
1930

FINDIF: A Software Package to Create Synthetic Seismograms by Finite Differences

by

Mary M. Hunt, Lee Gove,
and
Ralph A. Stephen

November 1983

Technical Report

Funding was provided by the Office of Naval Research under Contract N00014-79-C-0071; NR 083-004.

Approved for public release; distribution unlimited.

WOODS HOLE, MASSACHUSETTS 02543

WHOI-83-42

FINDIF: A Software Package to Create Synthetic Seismograms by Finite Differences

by

Mary M. Hunt, Lee Gove,
and
Ralph A. Stephen

Woods Hole Oceanographic Institution
Woods Hole, Massachusetts 02543

November 1983

Technical Report

Funding was provided by the Office of Naval Research under Contract N00014-79-C-0071; NR 083-004.

Approved for Distribution:

R. P. Von Herzen

Richard P. von Herzen, Chairman
Department of Geology and Geophysics

Abstract

In order to study seismic wave propagation through laterally varying sea floor structures, a software package has been created to generate synthetic seismograms by finite differences. The elastic wave equation can be solved in two dimensions either for point sources in cylindrical coordinates or for line sources in rectangular coordinates. Vertical and radial variations of the elastic parameters are allowed.

The package includes four programs. Input to the system consists of a short file containing parameter values to describe the model. The first program is used to initialize the system for the particular model being used. The source arrays and velocity matrices are each computed by a separate program. The final program, which actually carries out the finite difference calculations, includes six subroutines to implement different options based on alternative finite difference formulations. Two different kinds of output files are created by this program: one or more snap-shot files, and one time series file, which will usually include more than one series.

CONTENTS

I. Introduction

This report describes a software package used to generate synthetic seismograms by calculating the elastic wave equation in two dimensions by finite differences. The original finite difference code, which includes Kelly, Alterman, Ilan and Stephen formulations for cylindrical and rectangular co-ordinates with two different source configurations, was developed and written by R. A. S. between 1978 and 1983. The absorbing boundary subroutine was written by Mr. S. Emerman and is reported in Emerman and Stephen, 1983. Applications of the code and comparison of the results with the reflectivity method are discussed in Stephen (1983). L. G. assisted throughout the development by providing programming advice, writing input/output code and file handling procedures. L. G. also ran or directed a series of 'bench mark' tests for performance evaluation on the VAX, on the VAX with an array processor, and on the Cyber 205. The results of these tests are reported in the Appendix. M. M. H, commencing in the spring of 1983, has further modified the code to be more 'user friendly' and to be more easily transportable. M. M. H. also prepared this manual which is being written in order to facilitate the use of the software package by new users.

The program development is an on-going process; further enhancements are continually being made. The version documented and released here is a temporary stopping-place.

The program uses a variety of explicit finite difference techniques to solve the following equation:

$$\rho\ddot{\vec{u}} = (\lambda+\mu)\nabla(\nabla\cdot\vec{u})+\mu\nabla^2\vec{u} + \nabla\lambda(\nabla\cdot\vec{u})+\nabla\mu \times (\nabla\times\vec{u})+ 2\,(\nabla\mu\cdot\nabla)\vec{u}$$

where $\vec{u}$ is the particle displacement vector

λ,μ are Lamé's parameters, and

ρ is density.

The equation is solved in two dimensions either for point sources in cylindrical co-ordinates or for line sources in rectangular co-ordinates. Vertical and radial variations of the elastic parameters are allowed. The methods have been compared for accuracy with the reflectivity method by Stephen (1983).

II. The Model

The software depends heavily on the definition of the model, which is specified for the computer in the form of a model parameter file. A diagram of the model is given in Figure 1.

The model consists of three horizontal layers; the top layer is water, the middle layer is the sea-floor boundary layer, and the bottom layer is the sub-surface stratum. The top and bottom layers are assumed to be homogeneous, with a constant velocity. The boundary layer has a varying structure; the velocities within this layer are calculated by Program FDBNY. A grid is superimposed on the entire area of the model. The grid is divided into NN depth divisions, and MM distance (or range) divisions. The energy source is assumed to be located on the left edge of the model. The right edge and the bottom edge of the model are either rigid or absorbing boundaries.

There are two categories of parameters in the model parameter input file. Some parameters describe the physical model, and others describe the desired processing and output. A description of all parameters, in alphabetical order, is given in Table I. All times are in seconds, and distances are in kilometers.

Figure 1

DIAGRAM OF MODEL

RANGE-DEPTH CROSS SECTION: ISORB=0, IFLAT=0

```
        1                                                      MM
       1-----------------------------------------------------------
    ....1---------1                                              1
    .   1         1                                              1
    .   1A        1                                              1
 N. 1X            1                                              1
 S. ND1I          1                                              1
 W. 1S            1                                             R1
    .   1         1                   <DELR>                    I1
    .   1         1                  ^                          G1
    .   1         1                  D                          I1
    ....1----------                  E                          D1
        1                            L                           1
        1                            Z        VP1,VS1,RO1        1
        1                            v                           1
        1                                                        1
      NA1---------------------------------------------------------
        1                                                        1
        1O      VELOCITIES FOUND BY FDBNY                        1
        1F                                                       1
        1                                                        1
      NB1---------------------------------------------------------
        1                                                        1
        1S                                                      B1
        1Y                                                      N1
        1M                                                      D1
        1M                                                      Y1
        1E                                                      .1
        1T                           VP2,VS2,RO2                 1
        1R                                                       1
        1Y                                                       1
        1             RIGID BOUNDARY                             1
      NN1---------------------------------------------------------
```

Table I Description of variables in model parameter file

NAME	DATA TYPE	DESCRIPTION
DELR	REAL	range increment for each distance step - total distance = MM*DELR
DELT	REAL	time increment for each time step
DELZ	REAL	depth increment for each depth step - total depth = NN*DELZ
FILEID	CHARACTER*5	file identifier
IDENS	INTEGER	=1 for density variations in boundary layer, always = 1.
IEXPL	INTEGER	=1 for explicit formulation, =0 for implicit formulation (not implemented)
IFLAT	INTEGER	=1 for flat source region, =0 for square source region
IKELLY	INTEGER	=1 for Kelly source, =0 for Alterman source
IRECT	INTEGER	=1 for rectangular coordinates, =0 for cylindrical coordinates
ISORB	INTEGER	=1 for absorbing boundaries when ITRAN=0 or ITRAN=2, =0 for no absorbing boundaries
ITRAN	INTEGER	=0 for heterogeneous formulation in transition region =1 for Alterman explicit sharp boundary =2 for Ilan explicit sharp boundary =3 for gradient above Alterman explicit sharp boundary =4 for gradient above Ilan explicit sharp boundary
IVERT	INTEGER	controls contents of time series output =0 for pressure, =1 for vertical displacement, =2 for horizontal displacement
KINC	INTEGER	number of time steps between values in output time series
KK	INTEGER	value of final time step
KMARK	INTEGER	time step for first snap-shot file
KMINC	INTEGER	time step increment between snap-shot files
KOUTEN	INTEGER	last time step for output time series
KOUTST	INTEGER	first time step for output time series
KSTRT	INTEGER	first time step in execution
LABEL	INTEGER(20)	identification of parameter file - 80 characters
MINC	INTEGER	range increment for time series files
MM	INTEGER	total number of grid points in range (distance)
MOUTEN	INTEGER	final value of range for time series files
MOUTST	INTEGER	first value of range for time series files
MSW	INTEGER	width, in grid points, of source region (only used when IFLAT=0)
NA	INTEGER	depth grid value of upper edge of boundary layer
NB	INTEGER	depth grid value of lower edge of boundary layer
ND	INTEGER	height of source above top layer of model, when IFLAT=1 depth grid value of source, when IFLAT=0
NINC	INTEGER	depth increment for time series files
NN	INTEGER	total number of grid points in depth
NOUTEN	INTEGER	final value of depth for time series files
NOUTST	INTEGER	first value of depth for time series files
NSORCE	INTEGER	length of time series in FFT (2**NSORCE), when IKELLY=1
NSW	INTEGER	height, in grid points, of source region (only used when IFLAT=1)
PLSWID	REAL	width parameter, only used when IKELLY=1
RO1	REAL	density of upper layer
RO2	REAL	density of lower layer
ROT	REAL	density below/above interface
TSWAVE	REAL	time shift
VP1	REAL	P-wave velocity in upper layer
VP2	REAL	P-wave velocity in lower layer
VPT	REAL	P-wave velocity below/above interface
VS1	REAL	S-wave velocity in upper layer
VS2	REAL	S-wave velocity in lower layer
VST	REAL	S-wave velocity below-above interface

III. The Programs

The package includes four programs. Input to the system consists of a short file containing parameter values to describe the model, and to specify the amount of output desired. The final output includes two different kinds of files: one or more snap-shot files, and one time series file, which will usually include more than one series.

In an effort to minimize the computer resources used for a given model, minimum array sizes are computed and the specifications are stored in COMMON files. These files are then accessed during compilation by using INCLUDE statements in the programs. The charges incurred by running a model with larger than necessary arrays are usually greater than COMPILE and LINK charges.

The first program in the sequence, FDPREP, checks the model parameters for validity, and determines the correct dimensions for the arrays needed by the other programs. Two files are created with COMMON specifications correctly formatted for INCLUDE statements. Descriptions of the arrays in these COMMON blocks are given in Table II.

The second program, FDSORS, calculates sources. The program creates a file containing the model parameters which were used, and the source arrays computed.

The third program, a version of FDBNY, finds the matrices of P- and S-velocity squared, and the density matrix of the boundary layer for the model. This program also creates a file, containing model parameters used and the three matrices which were calculated.

The final program, FINDIF, uses the model parameter file, and the files produced by the other two programs, to create the final output. The snap-shot files give values of vertical displacement over the entire grid at the specified times. The time series give values of vertical displacement, horizontal displacement, or pressure at each of the selected grid points for the specified time interval.

Program FDSORS must be compiled using the specifications for COMMON block COMSOR which were found by FDPREP. Programs FDBNY and FINDIF must be compiled using the specifications for COMMON block COMFD8. All programs create a log file to give a permanent record of what was done. A diagram of the system is given in Figure 2.

Table II

Contents of COMMON block /COMFD8/

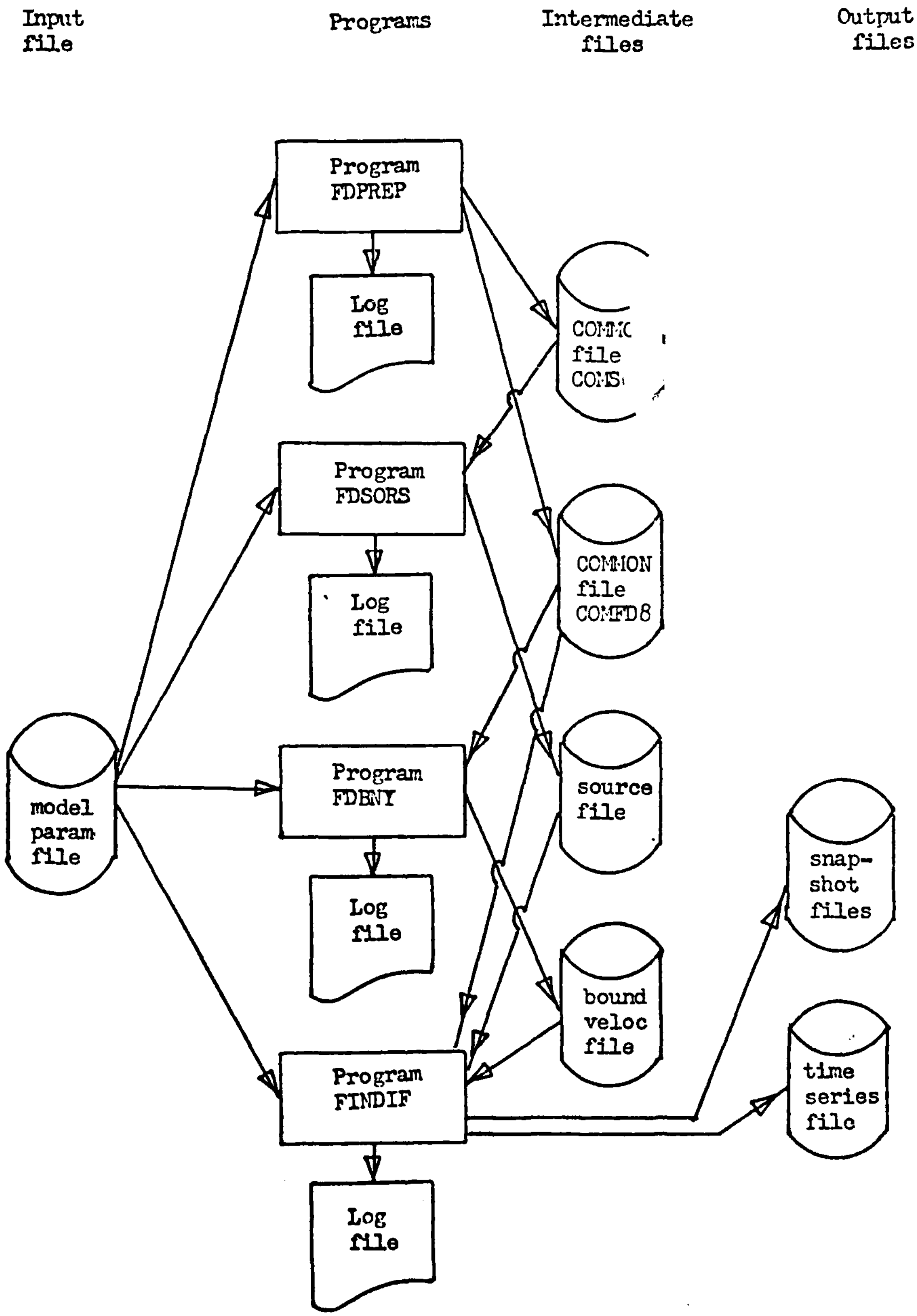
NAME	DIMENSIONS	DEFINITION
A	(MM,NN,2)	Horizontal displacement
B	(MM,NN,2)	Vertical displacement
AR	(MSW,NSW,2)	Used in computation,
BR	(MSW,NSW,2)	only when IFLAT = 0
C	(MM,2) or (MM)	These arrays are used in the computation,
D	(MM,2) or (MM)	only when ITRAN = 1.
E	(MM)	When IRECT = 1, the dimensions for C and D
F	(MM)	are (MM,2); otherwise (MM).
AN	(MM,2)	Used in computation,
BN	(MM)	only when ITRAN >= 1 and VS1 = 0.
RIGHTA	(MM)	Used in computation,
RIGHTB	(MM)	only when ISORB = 1
RIGAM1	(MM)	
RIGBM1	(MM)	
ROTA	(NN)	
ROTB	(NN)	
ROTAM1	(NN)	
ROTBM1	(NN)	
ADATA	(NMT,NNT,NKT)	Time series array; NMT=(MOUTEN-MOUTST)/MINC+1, etc
SORSA	(MDIM)	Source arrays - MDIM = MM if IFLAT = 1;
SORSB	(MDIM)	else MDIM = 4*MSW + 2*NSW - 8
VP32	(MM,NBNDY)	P-wave velocity squared in boundary layer
VS32	(MM,NBNDY)	S-wave velocity squared in boundary layer
RO3	(MM,NBNDY)	Density in boundary layer
		NBNDY = NB-NA+2 if ITRAN = 1,2;
		else NBNDY = NB-NA+3.

Contents of COMMON block /COMSOR/

NAME	DIMENSIONS	DEFINITION
BSERIES	(2**NSORCE)	Complex array
DUM1	(KDIM,MDIM)	Source arrays - KDIM = KK - KSTRT + 1
DUM2	(KDIM,MDIM)	MDIM = MM if IFLAT = 1
		else MDIM = 4*MSW + 2*NSW - 8

Figure 2

SYSTEM DIAGRAM



IV. The Files

A. Model Parameter File

This is a sequential formatted file, most of which is read with list-directed READ statements (free-field input). The Fortran unit number is 55. The record contents follow.

1. Record 1 consists of up to 80 characters containing identification for the model.

2. Record 2 contains, in the first 5 characters, identification used in the names of input and output files.

3. Record 3 (free-field) contains values of:

   IRECT, IDENS, IFLAT, IKELLY, IEXPL, IVERT, ITRAN, ISORB

4. Record 4 (free-field) contains values of:

   MM, NN, KK, KSTRT, DELT, DELR, DELZ

5. Record 5 (free-field) contains values of:

   KOUTST, KOUTEN, MOUTST, MOUTEN, NOUTST, NOUTEN

6. Record 6 (free-field) contains values of:

   KINC, MINC, NINC, KMARK, KMINC

7. Record 7 (free-field) contains values of:

   VP1, VS1, RO1, VP2, VS2, RO2, NA, NB, VPT, VST, ROT

8. Record 8 (free-field) contains values of:

   ND, MSW, NSW

9. Record 9 (free-field) contains values of:

   NSORCE, PLSWID, TSWAVE

B. Boundary velocity file

This file is created by FDBNY, and is read by FINDIF. It is a sequential, unformatted file, created with binary WRITE statements. The Fortran unit number is 54. The records are:

1. Record 1 contains values of:

   IEXPL, IDENS, MM, NA, NB

2. Record 2 contains values of:

   VP1, VS1, RO1, VP2, VS2, RO2

3. Record 3 contains all values of array VP32. The first dimension of this array is the value of MM. The second dimension is the value of

   NBNDY = NB - NA + 2 when ITRAN = 1 or ITRAN = 2

   = NB - NA + 3 otherwise

4. Record 4 contains all values of array VS32. Dimensions are the same as VP32.

5. Record 5 is used only when IDENS = 1. It contains all values of array RO3. The dimensions are the same as VP32.

C. Source File

This file is created by FDSORS, and is read by FINDIF. It is a sequential, unformatted file, created with binary WRITE statements. The Fortran unit number is 50. The records are:

1. Record 1 contains values of:

   IEXPL, IFLAT, IKELLY, VP1, KSTRT, KK, IRECT

2. Record 2 contains values of:

   ND, MSW, NSW, DELT, DELZ, DELR, NSORCE, PLSWID, TSWAVE

3. Record 3 contains values of SORSA for the first time step. The dimension of SORSA is:

   MM if IFLAT = 1

   4*MSW + 2*NSW - 8 if IFLAT = 0

4. Record 4 contains values of SORSB for the first time step. The dimension is the same as for SORSA.

Records 3 and 4 are repeated for each time step. Total number of these two records is KK - KSTRT + 1.

D. Snap-shot Files

Snap-shot files are created by FINDIF. They are sequential, unformatted files, created with binary WRITE statements. The Fortran unit number is 3. The first snap-shot file is created at

time step KMARK. Thereafter, a snap-shot file is created every KMINC timesteps, until the final timestep (KK) is reached. The names of these files are created by the program. The first 5 characters are the FILEID. The next 4 characters are the numeric value of the timestep. The file extent is .SNS. The contents of these files follow:

1. The first record contains values of all integer variables in COMMON block MODPAR. These variables are:

   LABEL(20), IRECT, IDENS, IFLAT, IKELLY, IEXPL, IVERT, ITRAN, ISORB, NA, NB, ND, NSW, MSW, MM, NN, KK, KKSORS, KOUTST, KOUTEN, KINC, NOUTST, NOUTEN, NINC, MOUTST, MOUTEN, MINC, KSTRT, KMARK, KMINC, NSORCE, KCNT, NSAMP

2. The second record contains values of all real variables in COMMON block MODPAR. These variables are:

   VP1, VP2, VPT, VS1, VS2, VST, RO1, RO2, ROT, DELT, DELR, DELZ, PLSWID, TSWAVE, TIME

3. The third and last record contains values of vertical displacement over the entire grid (MM * NN values).

E. Time Series File

This file is created by Program FINDIF. It is a sequential, unformatted file, created with binary WRITE statements. The Fortran unit number is 2. The name of the file is the same as FILEID, with an extent of .TST. The contents of the time series file are controlled by the value of IVERT. If IVERT = 0, the file will contain values of pressure; if IVERT = 1, it will contain values of vertical displacement; if IVERT = 2, it will contain values of horizontal displacement.

The number of individual time series is controlled by values of MOUTST, MOUTEN, MINC, and NOUTST, NOUTEN, and NINC. One series will be generated for every MINC'th range grid between MOUTST and MOUTEN, and every NINC'th depth grid between NOUTST and NOUTEN. Each time series will include every KINC'th time step between KOUTST and KOUTEN.

The contents of the records in the file are:

1. The first record contains values of all integer variables in COMMON block MODPAR. These variables are listed in the snap-shot file description.

2. The second record contains values of all real variables in COMMON block MODPAR, listed in the snap-shot file description.

3. The third record contains the following:

LABEL(I),I=1,5	first 20 characters of label
MLOC	range grid value of series
NLOC	depth grid value of series
KCNT	number of values in series
NSAMP	number of samples per second in series
RANGE	horizontal distance
TIME	time of first value, in seconds
ADMAX	maximum absolute value in series

4. This record contains the time series described in record 3. Record types 3 and 4 are repeated for as many time series as requested.

V. The subprograms

A number of subprograms are included. These are listed, with brief descriptions, in Table III. It will be noted that there are 6 different TSTEP routines. The one used depends on the values of IRECT, VS1, and ITRAN, according to the table below.

IRECT	VS1	ITRAN	TSTEP
1	$\neq$ 0.0	0	1
1	$\neq$ 0.0	1	1
1	$\neq$ 0.0	2	1
1	= 0.0	0	1
1	= 0.0	2	3
0	$\neq$ 0.0	0	2
0	$\neq$ 0.0	1	2
0	$\neq$ 0.0	3	5
0	$\neq$ 0.0	4	6
0	= 0.0	0	2
0	= 0.0	1	4
0	= 0.0	2	4
0	= 0.0	3	5
0	= 0.0	4	6

Combinations other than these have not been implemented.

Table III

Subprograms used.

NAME	Used by	Variable COMMON needed	Function
LOGOUT	all	none	Creates first part of Log file
RDMPAR	all	none	Reads model parameter file
DATIM	all	none	Access current date and time from operating system, and writes them.
OPNCOM	FDPREP	none	Opens files for COMMON specifications
BESJ	FDSORS	none	Computes Bessel function values J0 or J1
BESY	FDSORS	none	Computes Bessel function values Y0 or Y1
COOL	FDSORS	none	Does FFT
KELLY7	FDSORS	COMSOR	Finds displacement due to Kelly source
SOURCE	FDSORS	none	Finds displacement due to Alterman source
ABSORB	FINDIF	COMFD8	Does absorbtion calculation
CONIT	FINDIF	COMFD8	Initializes variables and arrays
FSETUP	FINDIF	none	Opens files
SNPOUT	FINDIF	none	Outputs snap-shot file.
TSOUT	FINDIF	none	Outputs time series file.
TIMIT TIMSHO	FINDIF	none	Initializes system timer Entry to TIMIT, finds elapsed time.
TSTEP1 TSTEP2 TSTEP3 TSTEP4 TSTEP5 TSTEP6	FINDIF	COMFD8	These routines compute the displacements for one time-step.

VI. Implementation Notes

The programs have been written in Fortran 77 for a VAX 11/780 under VMS. An effort has been made to avoid non-standard features. Specifically, all identifiers are 6 characters or less, and the DO WHILE feature is not used. The programs all use the BLOCK IF capability, and do use character variables. If the INCLUDE statement is not available, programs and some subroutines will have to be edited to change some of the array dimensions. The various TSTEP routines are rather long, so we have created a subroutine library which includes dummy versions of them. When we load FINDIF, we specifically include the TSTEP needed, and satisfy the unused references to the other TSTEP routines from the library. It may be necessary at some installations to use some other technique.

An effort has been made to isolate all input and output, and all non-standard features, in subroutines. The following routines might need changes:

TIMIT to initialize the system timer
TIMSHO, an ENTRY to TIMIT, finds elapsed CPU time

FSETUP opens output Log file, source file, and boundary velocity file

SNPOUT creates name of snap-shot file, opens file, writes data in file, and closes file

TSOUT creates name of time series file, opens file, writes data in file, and closes file

RDMPAR used by all programs, reads the model parameter file. It uses list-directed READ statements.

DATIM used by all programs, gets current date and time from operating system.

OPNCOM used by FDPREP, opens files for new COMMON specifications.

VII. Usage on the VAX

A. General

This section describes how the system is set up to run on the VAX here at WHOI. Changes might be needed for other computers. In general, ASSIGN commands are required for input files, and output file names are created by the programs using the FILEID from the model parameter file with a different file extent (or file type) for each output file.

B. Model Parameter File

The first step is to create a model parameter file. The contents of the records are listed in Section IV A, and descriptions of the variables are in Table I. All records except the first two are in free-field format, so spacing is not important. Let us, in this example, call this file MODEL.PAR, and let us use MODL1 as the file identifier.

C. Run FDPREP

Next, Program FDPREP should be run. This program does the following:

1. Checks model parameters for consistency
2. Creates two COMMON block files:

   COMFD8.FOR used by FDBNY and FINDIF

   COMSOR.FOR used by FDSORS.
3. Creates a Log file summarizing its activities. In this case, the file will be MODL1.LG1

Before running FDPREP, you must use an ASSIGN command to connect the model parameter file to unit 55. Commands to run FDPREP are

```
$ASSIGN MODEL.PAR FOR055

$RUN FDPREP
```

D. Run FDBNY

Now Program FDBNY can be compiled and run. Here again, the only ASSIGN command required is for the model parameter file. The program reads this file and creates the following two files:

1. The file defining the velocity structure in the transition zone. The name of this file will be MODL1.BNY

2. A log file, named MODL1.LG2

The commands to compile and run FDBNY are:

```
$FORTRAN FDBNY
$LINK FDBNY,FDLIB/LIB
$ASSIGN MODEL.PAR FOR055
$RUN FDBNY
```

E. Run FDSORS

We can now compile and run Program FDSORS. This is similar to the procedure for FDBNY. The files created are:

1. A file defining the energy source for each time step. The name of this file will be MODL1.SOR

2. A log file, named MODL1.LG3

The commands to compile and run FDSORS are:

```
$FORTRAN FDSORS
$LINK  FDSORS,FDLIB/LIB
$ASSIGN MODEL.PAR  FOR055
$RUN FDSORS
```

Subroutine KELLY7 is in the same file as FDSORS.

F. Run FINDIF

Finally, we can compile and run FINDIF. We must also compile the version of TSTEP which will be needed. This information is included in the log file created by FDPREP. The commands to run FINDIF are:

```
$FORTRAN FINDIF
$FORTRAN TSTEPn          use the selected version
$FORTRAN FINSUB          this file includes ABSORB and CONIT
$LINK FINDIF,TSTEPn,FINSUB, FDLIB/LIB
$ASSIGN MODEL.PAR FOR055
$ASSIGN MODL1.BNY FOR054
$ASSIGN MODL1.SOR FOR050
$RUN FINDIF
```

The files created by FINDIF are:

1. Log file, named MODL1.LG4

2. A time series file, named MODL1.TST

3. One or more snap-shot files. The names includes the time step. For example, a snap-shot file of time step 1000 will be MODL11000.SNS

G. Boundary and Source Files

It is not necessary to create new boundary and source files for each run of FINDIF. However, there are some parameters which must be the same as those used to create the files. For FDBNY, these parameters are MM, NA, NB, VP1, VS1, RO1, VP2, VS2, and RO2. For FDSORS, they are IFLAT, IKELLY, VP1, KSTRT, KK, ND, MSW, IRECT, DELT, DELZ, DELR, NSORCE, PLSWID, and TSWAVE. If any of these parameters in FINDIF is not the same as that used to create the boundary and/or source files, an error message is output, and FINDIF terminates.

VIII. Error and Diagnostic Messages

The following error messages are included so the conditions causing the errors can be avoided.

A. FDPREP

1. The first two messages are the result of inconsistencies in the time step parameters. KSTRT and KK are the first and last time steps. KOUTST and KMARK are the first time step for time series and snap-shot files respectively. KOUTEN is the time step for the end of the time series. The messages are:

   KOUTST =___ or KMARK =___ IS LESS THAN KSTRT =___

   KOUTEN =___ IS GREATER THAN KK =___

2. When IFLAT = 0, NSW must be an odd number. If not, the following message will be output

   NSW IS NOT AN ODD NUMBER - JOB STOPPED

3. Invalid combinations of parameters can produce the following:

   INVALID COMBINATION: ISORB =___ ITRAN =___

   INVALID COMBINATION: IRECT =___ VS1 =___ITRAN =___

4. The last is a warning. The program will continue.

   WARNING - ABSORBING BOUNDARIES NOT CODED FOR THIS COMBINATION.

B. FINDIF

1. The program will not run if the parameters used to create the boundary and/or source files do not match the current parameters. The messages are:

   BOUNDARY PARAMETERS DO NOT MATCH

   SOURCE PARAMETERS DO NOT MATCH

   In each case, the offending parameter values are output.

2. The next message can appear only when the model parameter file is changed after running FDPREP, or the user does something stupid.

   INVALID COMBINATION OF PARAMETERS: IRECT =___ ITRAN =___ VS1 =___

3. File errors:

ERROR____ OPENING FILE_____

SNAPSHOT FILE ERROR. CODE =_____ KMARK =_____

ERROR ON TIME SERIES FILE. CODE =_____

As can be seen, in each case the error code is included in the message.

IX. Sample Runs

A group of sample runs was made, using the same basic parameters; the only changes in the parameter file were those required by the use of different options. The contents of the first model parameter file are listed with the output of the first run. The changes to create the other parameter files are given in Table IV. The sample run of the first model follows, step by step, the instructions in Section VII. Simple contour plots of the snap-shot files and some time series plots are included.

It will be noticed that the snap-shots for STEVB and STEVD (cylindrical coordinates and a square source) include some anomalies in the source area. This noise is a result of the small size of the model and is not present in larger models.

A. Model Parameter File

Below is a listing of the model parameter file used in the first run. As can be seen, the name of the file is STEV1.PAR, and the FILEID is also STEV1. These two names do not have to be the same, although it saves some confusion if they are.

```
STEV1   FINAL TEST FOR RELEASE
STEV1
 1, 1, 1, 1, 1, 1, 0, 1
 72, 72, 300, 1, 0.008, 0.04, 0.04
 1, 300, 1, 72, 37, 37
 2, 5, 1, 100, 100
 1.5, 0.0, 1.0, 2.5, 1.4, 1.5, 17, 47, 0.0, 0.0, 0.0
 -11, 0, 0
 10, 50.0, 0.5292
```

Table IV

Parameter Values for Test Runs

Parameter file	STEV1	STEV2	STEV3	STEV4	STEV5	STEVA	STEVB	STEVC	STEVD
TSTEP routine	1	1	2	3	4	1	2	3	4
IFLAT	1	1	1	1	1	0	0	0	0
ND	-11	-11	-11	-11	-11	10	10	10	10
MSW,NSW	0,0	0,0	0,0	0,0	0,0	5,11	5,11	5,11	5,11
ITRAN	0	0	0	2	2	0	0	2	2
NA,NB	17,47	17,47	17,47	30,30	30,30	17,47	17,47	30,30	30,30
IRECT	1	1	0	1	0	1	0	1	0
ISORB	1	0	0	1	0	1	0	0	0
Boundary file	STEV1	STEV1	STEV1	STEV4	STEV4	STEV1	STEV1	STEV4	STEV4
Source file	STEV1	STEV1	STEV3	STEV1	STEV3	STEVA	STEVB	STEVA	STEVB
FINDIF CPU time (Minutes)	5:50	5:41	6:38	2:52	3:43	6:15	6:35	2:46	3:42

B. Run Program FDPREP

A listing of the log file is included

```
$ ASSIGN STEV1.PAR FOR055
$ RUN FDPREP
FDPREP COMPLETED--NO ERRORS
$ TYPE STEV1.LG1


    PROGRAM FDPREP                   VERSION OF  20-SEP-83
                                    DATE AND TIME OF RUN  23-SEP-83   13:50:48

    FILE ID IS STEV1
  VALUES OF INPUT PARAMETERS:

IRECT =  1    IDENS =  1    IFLAT =  1    IKELLY =  1    IEXPL =  1    IVERT =  1
ITRAN =  0    ISORB =  1
MM =     72   NN =     72   KK =    300   KSTRT =     1
NA =    17    NB =    47    ND =    -11   MSW =      0    NSW =      0
DELT =  0.008               DELR =  0.040                DELZ =  0.040

OUTPUT SPANS AND INCREMENTS:

   TIMES        1      300       2
   RANGES       1       72       5
   DEPTHS      37       37       1
   SNAPS      100              100

VELOCITIES:

VP1 =  1.500   VS1 =  0.000   RO1 =  1.000
VP2 =  2.500   VS2 =  1.400   RO2 =  1.500
VPT =  0.000   VST =  0.000   ROT =  0.000

KELLY SOURCE PARAMETERS:

NSORCE =   10      PLSWID =     50.0000      TSWAVE =      0.3292


 FINDIF WILL USE SUBROUTINE FSTEP1
    NA INTERCEPTED AT TIME STEP      93
    NN INTERCEPTED NO EARLIER THAN TIME STEP:     101
    MM INTERCEPTED NO EARLIER THAN TIME STEP:     101

 DIMENSION FOR ARRAYS
 FOR A AND B:     72     72      2
 FOR AR AND BR:      1      1      2
 FOR C, D: (  72,2)
 FOR AN: ( 72,2)    BN: ( 72)
 FOR RIGHTA,RIGHTB,RIGAM1,RIGBM1:     72
 FOR BOTA,BOTB,BOTAM1,BOTBM1:     72
 FOR ADATA:     15      1    100
 FOR SIRSA AND SORCE:     72
 FOR VP32 AND VS32:     72     33
 FOR RO3:     72     33
```

C. Compile, link and run FDBNY

Since this was done immediately after running FDPREP, we did not need to repeat the ASSIGN command for STEV1.PAR. A partial listing of the log file is included.

```
$ FORTRAN FDBNY
$ LINK FDBNY,FDLIB/LIB
$ RUN FDBNY
FORTRAN STOP
$ TYPE STEV1.LG2


    PROGRAM FDBNY                    VERSION OF    9-JUL-83
                                   DATE AND TIME OF RUN  23-SEP-83   13:51:51
    FILE ID IS STEV1
  VALUES OF INPUT PARAMETERS:

IRECT =  1   IDENS =  1   IFLAT =  1   IKELLY =  1   IEXPL =  1   IVERT =  1
ITRAN =  0   ISORB =  1
MM =    72   NN =    72   KK =    300  KSTRT =     1
NA =   17    NB =    47   ND =    -11  MSW =       0   NSW =       0
DELT =  0.008              DELR =  0.040               DELZ =  0.040


VELOCITIES:

VP1 =  1.500   VS1 =  0.000   RO1 =  1.000
VP2 =  2.500   VS2 =  1.400   RO2 =  1.500
VPT =  0.000   VST =  0.000   ROT =  0.000
VP**2 IN TRANSITION ZONE, (I,J), I =  1,  72,  8 J =  1,  33
  2.250  2.250  2.250  2.250  2.250  2.250  2.250  2.250  2.250  2.441
  2.441  2.441  2.441  2.441  2.441  2.441  2.441  2.441  2.540  2.540
  2.540  2.540  2.540  2.540  2.540  2.540  2.540  2.641  2.641  2.641
  2.641  2.641  2.641  2.641  2.641  2.641  2.743  2.743  2.743  2.743
  2.743  2.743  2.743  2.743  2.743  2.848  2.848  2.848  2.848  2.848
  2.848  2.848  2.848  2.843  2.954  2.754  2.954  2.954  2.954  2.954
  2.954  2.954  2.954  3.063  3.063  3.063  3.063  3.063  3.063  3.063
  3.063  3.063  3.173  3.173  3.173  3.173  3.173  3.173  3.173  3.173
  3.173  3.285  3.285  3.285  3.285  3.285  3.285  3.285  3.285  3.285
  3.399  3.399  3.397  3.399  3.399  3.399  3.397  3.399  3.399  3.516
  3.516  3.516  3.516  3.516  3.516  3.516  3.516  3.516  3.634  3.634
  3.634  3.634  3.634  3.634  3.634  3.634  3.634  3.754  3.754  3.754
  3.754  3.754  3.754  3.754  3.754  3.754  3.876  3.876  3.876  3.876
  3.876  3.876  3.876  3.876  3.876  4.000  4.000  4.000  4.000  4.000
  4.000  4.000  4.000  4.000  4.126  4.126  4.126  4.126  4.126  4.126
  4.126  4.126  4.126  4.254  1.254  4.254  4.254  4.254  4.254  4.254
  4.254  4.254  4.384  4.384  4.384  4.384  4.384  4.384  4.384  4.384
  4.384  4.516  4.516  4.516  4.516  4.516  4.516  4.516  4.516  4.516
  4.649  4.649  4.649  4.649  4.649  4.649  4.649  4.649  4.649  4.785
  4.785  4.785  4.785  4.785  4.785  4.785  4.785  4.785  4.923  4.923
  4.923  4.923  4.923  4.923  4.923  4.923  4.923  5.063  5.063  5.063
  5.063  5.063  5.063  5.063  5.063  5.063  5.204  5.204  5.204  5.204
  5.204  5.204  5.204  5.204  5.204  5.348  5.348  5.348  5.348  5.348
```

D. Compile, link, and run FDSORS

Again, we have included a listing of the log file.

```
$ FORTRAN FDSORS
$ LINK FDSORS,FDLIB/LIB
$ RUN FDSORS
FORTRAN STOP
$ TYPE STEV1.LG3


    PROGRAM FDSORS                     VERSION OF  12-SEP-83
                                     DATE AND TIME OF RUN  23-SEP-83    13:52:56
    FILE ID IS STEV1
  VALUES OF INPUT PARAMETERS:

IRECT =  1    IDENS =  1    IFLAT =  1    IKELLY =  1    IEXPL =  1    IVERT =  1
ITRAN =  0    ISORB =  1
MM =     72   NN =     72   KK =     300   KSTRT =     1
NA =    17    NB =    47    ND =    -11    MSW =       0    NSW =   . 0
DELT =  0.008               DELR =   0.040                  DELZ =   0.040

OUTPUT SPANS AND INCREMENTS:

   TIMES        1      300        2
   RANGES       1       72        5
   DEPTHS      37       37        1
   SNAPS      100               100

VELOCITIES:

VP1 =  1.500   VS1 =  0.000   RO1 =  1.000
VP2 =  2.500   VS2 =  1.400   RO2 =  1.500
VPT =  0.000   VST =  0.000   ROT =  0.000

KELLY SOURCE PARAMETERS:

NSORCE =   10      PLSWID =     50.0000      TSWAVE =      0.5292
```

E. Run FINDIF

Finally, we compile, link, and run FINDIF. We use TSTEP1 because Program FDPREP informed us which TSTEP routine would be needed. As can be seen, Program FINDIF displays the elapsed time and CPU time every 50 time steps. The log file is included on the next page.

```
$ FORTRAN FINDIF
$ FORTRAN TSTEP1
$ FORTRAN FINSUB
$ ASSIGN STEV1.SOR FOR050
$ ASSIGN STEV1.BNY FOR054
$ LINK FINDIF,TSTEP1,FINSUB,FDLIB/LIB
$ RUN FINDIF
 PERFORMANCE STATISTICS AT TIME STEP   50
 ELAPSED: 00:01:08.53  CPU: 0:01:01.08  BUFIO: 2  DIRIO: 3  FAULTS: 59

 PERFORMANCE STATISTICS AT TIME STEP  100
 ELAPSED: 00:02:18.00  CPU: 0:02:03.30  BUFIO: 8  DIRIO: 10  FAULTS: 109

 PERFORMANCE STATISTICS AT TIME STEP  150
 ELAPSED: 00:03:27.43  CPU: 0:03:05.32  BUFIO: 11  DIRIO: 13  FAULTS: 261

 PERFORMANCE STATISTICS AT TIME STEP  200
 ELAPSED: 00:04:48.84  CPU: 0:04:07.57  BUFIO: 17  DIRIO: 20  FAULTS: 280

 PERFORMANCE STATISTICS AT TIME STEP  250
 ELAPSED: 00:06:20.06  CPU: 0:05:10.19  BUFIO: 20  DIRIO: 23  FAULTS: 283

 PERFORMANCE STATISTICS AT TIME STEP  300
 ELAPSED: 00:07:52.30  CPU: 0:06:12.78  BUFIO: 26  DIRIO: 29  FAULTS: 285
FORTRAN STOP
```

```
$ TYPE STEV1.LG4


   PROGRAM FINDIF                     VERSION OF  22-AUG 83
                                     DATE AND TIME OF RUN  23-SEP-83   13:59:56
   FILE ID IS STEV1
  VALUES OF INPUT PARAMETERS:

IRECT =  1   IDENS =  1   IFLAT =  1   IKELLY =  1   IEXFL =  1   IVERT =  1
ITRAN =  0   ISORD =  1
MM =    72   NN =    72   KK =   300   KSTRT =    1
NA =   17    NB =   47    ND =   -11   MSW =     0   NSW =     0
DELT =  0.008             DELR =  0.040              DELZ =  0.040

OUTPUT SPANS AND INCREMENTS:

   TIMES        1      300        2
   RANGES       1       72        5
   DEPTHS      37       37        1
   SNAPS      100               100

VELOCITIES:

VP1 =  1.500   VS1 =  0.000   RO1 =  1.000
VP2 =  2.500   VS2 =  1.400   RO2 =  1.500
VPT =  0.000   VST =  0.000   ROT =  0.000

KELLY SOURCE PARAMETERS:

NSORCE =   10      PLSWID =    50.0000      TSWAVE =     0.5292


  2.250  2.441  2.540  2.641  2.743  2.848
  2.954  3.063  3.173  3.285  3.399  3.516
  3.634  3.754  3.876  4.000  4.126  4.254
  4.384  4.516  4.649  4.785  4.923  5.063
  5.204  5.348  5.493  5.641  5.790  5.941
  6.095  6.250  6.250
  0.000  0.008  0.017  0.031  0.048  0.069
  0.094  0.122  0.155  0.191  0.232  0.276
  0.323  0.375  0.431  0.490  0.553  0.620
  0.691  0.766  0.844  0.926  1.013  1.102
  1.196  1.294  1.395  1.501  1.610  1.723
  1.839  1.960  1.960
  1.000  1.031  1.047  1.063  1.078  1.094
  1.109  1.125  1.141  1.156  1.172  1.188
  1.203  1.219  1.234  1.250  1.266  1.231
  1.297  1.313  1.328  1.344  1.359  1.375
  1.391  1.406  1.422  1.438  1.453  1.469
  1.484  1.500  1.500
VP**2 IN TRANSITION ZONE, (I,J),I=1,     72     8 J=1,     33
  2.250  2.250  2.250  2.250  2.250  2.250  2.250  2.250  2.250  2.441
  2.441  2.441  2.441  2.441  2.441  2.441  2.441  2.441  2.540  2.540
  2.540  2.540  2.540  2.540  2.540  2.540  2.540  2.641  2.641  2.641
  2.641  2.641  2.641  2.641  2.641  2.641  2.743  2.743  2.743  2.743
  2.743  2.743  2.743  2.743  2.743  2.848  2.848  2.848  2.848  2.848
  2.848  2.848  2.848  2.848  2.954  2.954  2.954  2.954  2.954  2.954
```

The following pages contain contour plots of the snap-shot files. A few words of explanation about the format of these plots will help in understanding them. The FILEID and time step number are in the lower right corner of each plot. The number in the lower left corner is the scaling factor used for the plot, which is selected by the plot program. In some cases, we have used the same scaling for all timesteps, and in others, we have used a different scale for each timestep. The dashed lines indicate negative contours.

Figure 3

Contour plots of snap-shot files from file STEV1

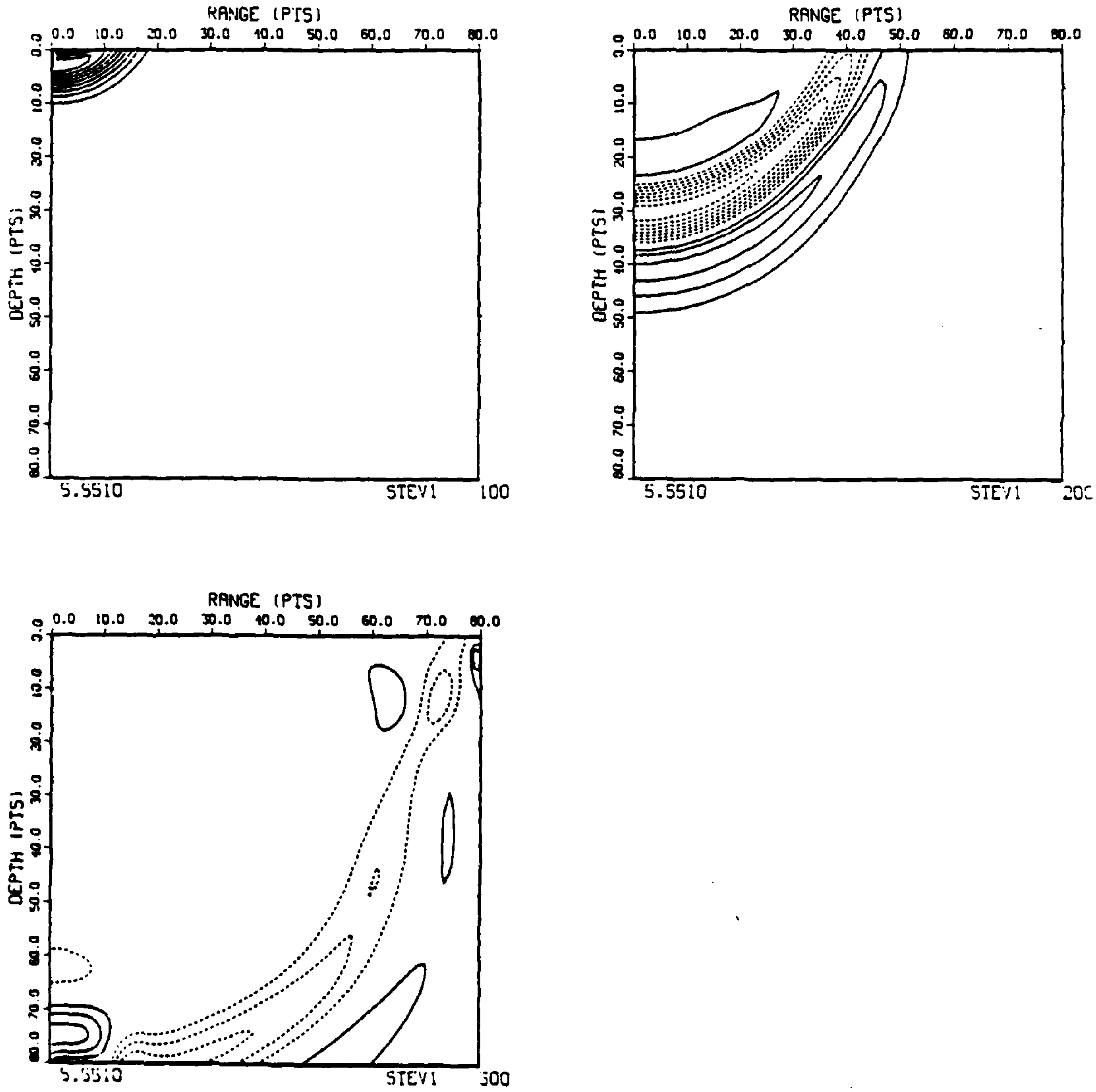


Figure 4

Contour plots of snap-shot files from model STEV2

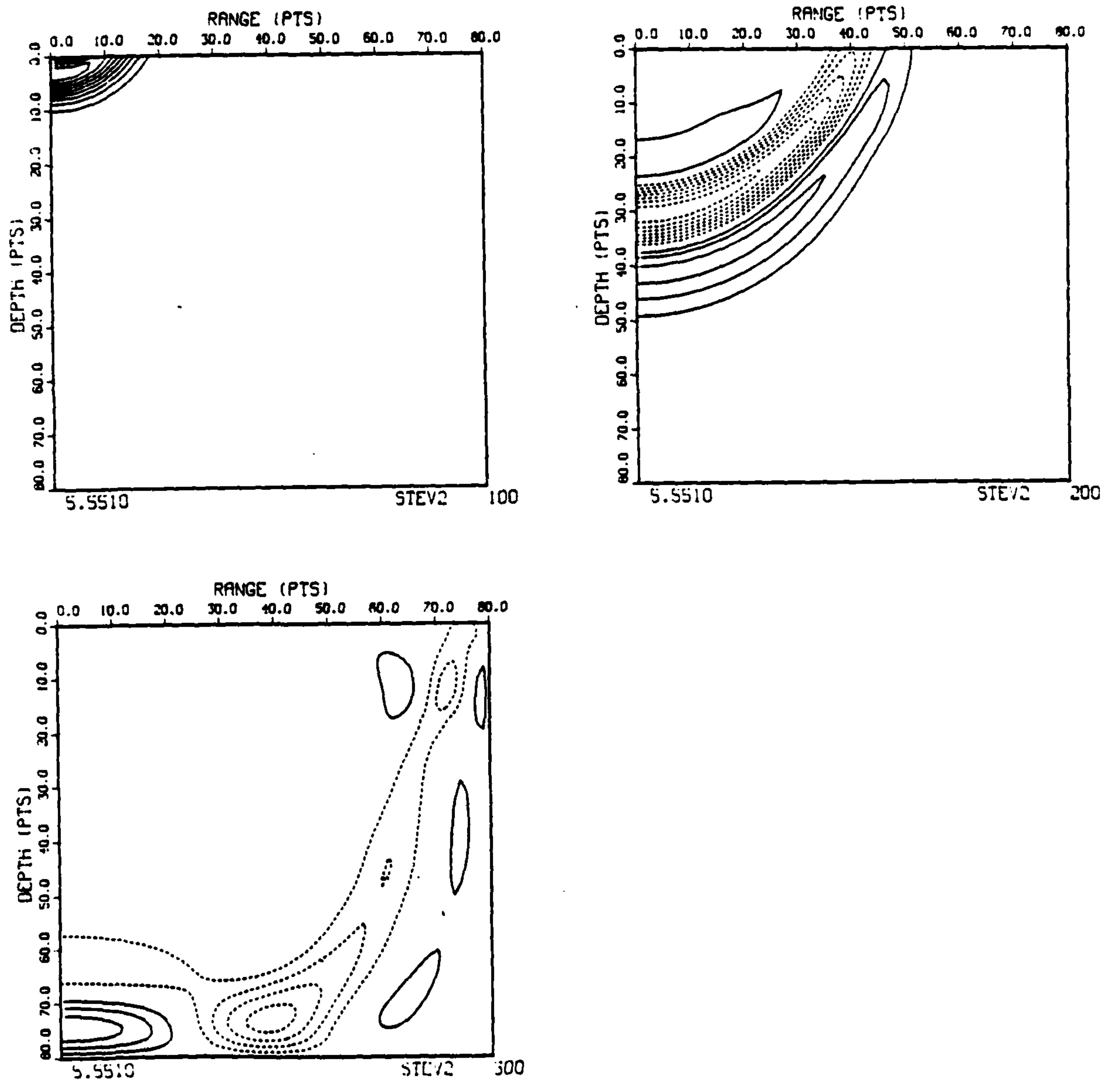


Figure 5

Contour plots of snap-shot files from model STEV3

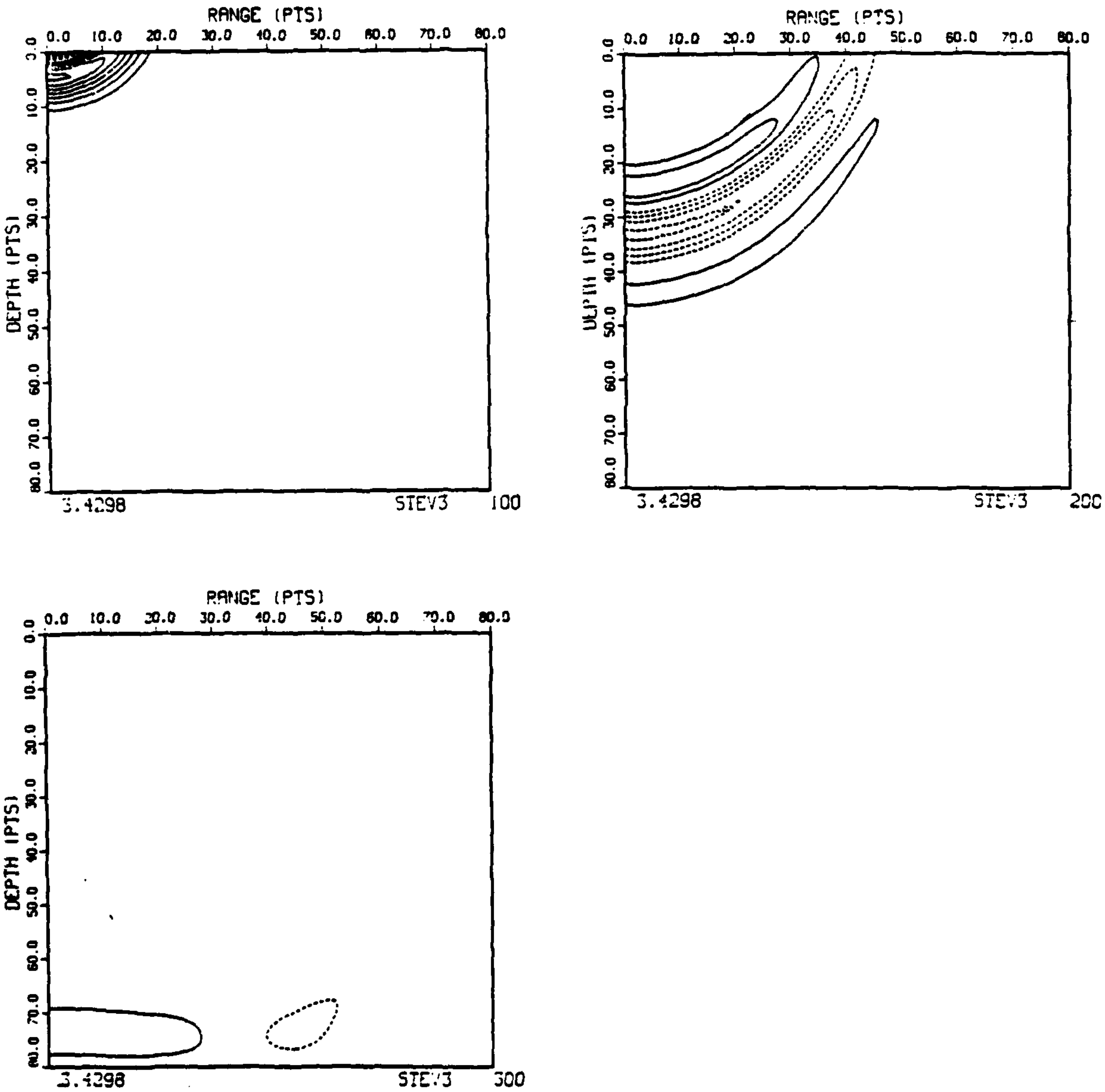


Figure 6

Contour plots of snap-shot files from model STEV4

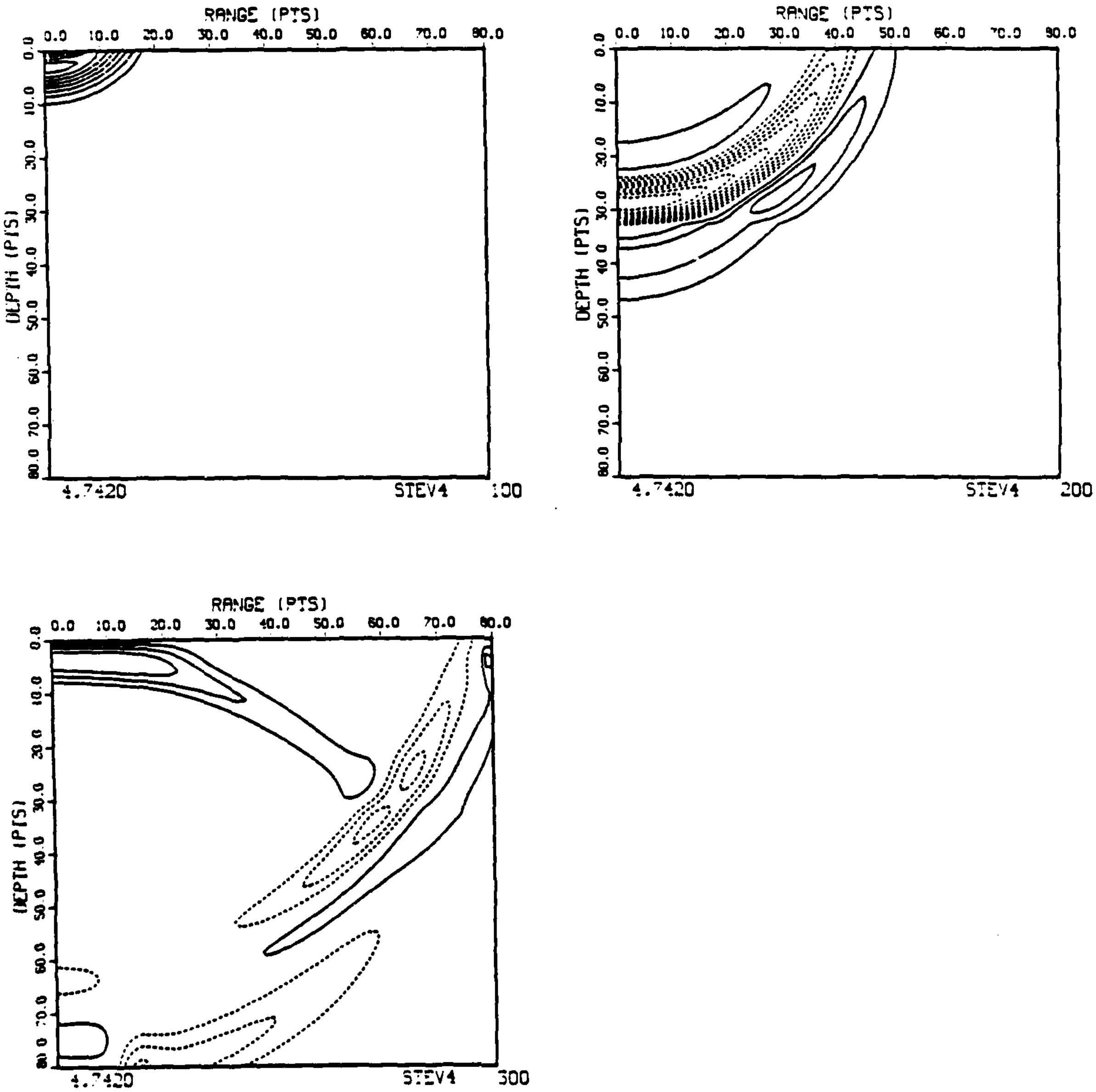


Figure 7

Contour plots of snap-shot files from model STEV5 .

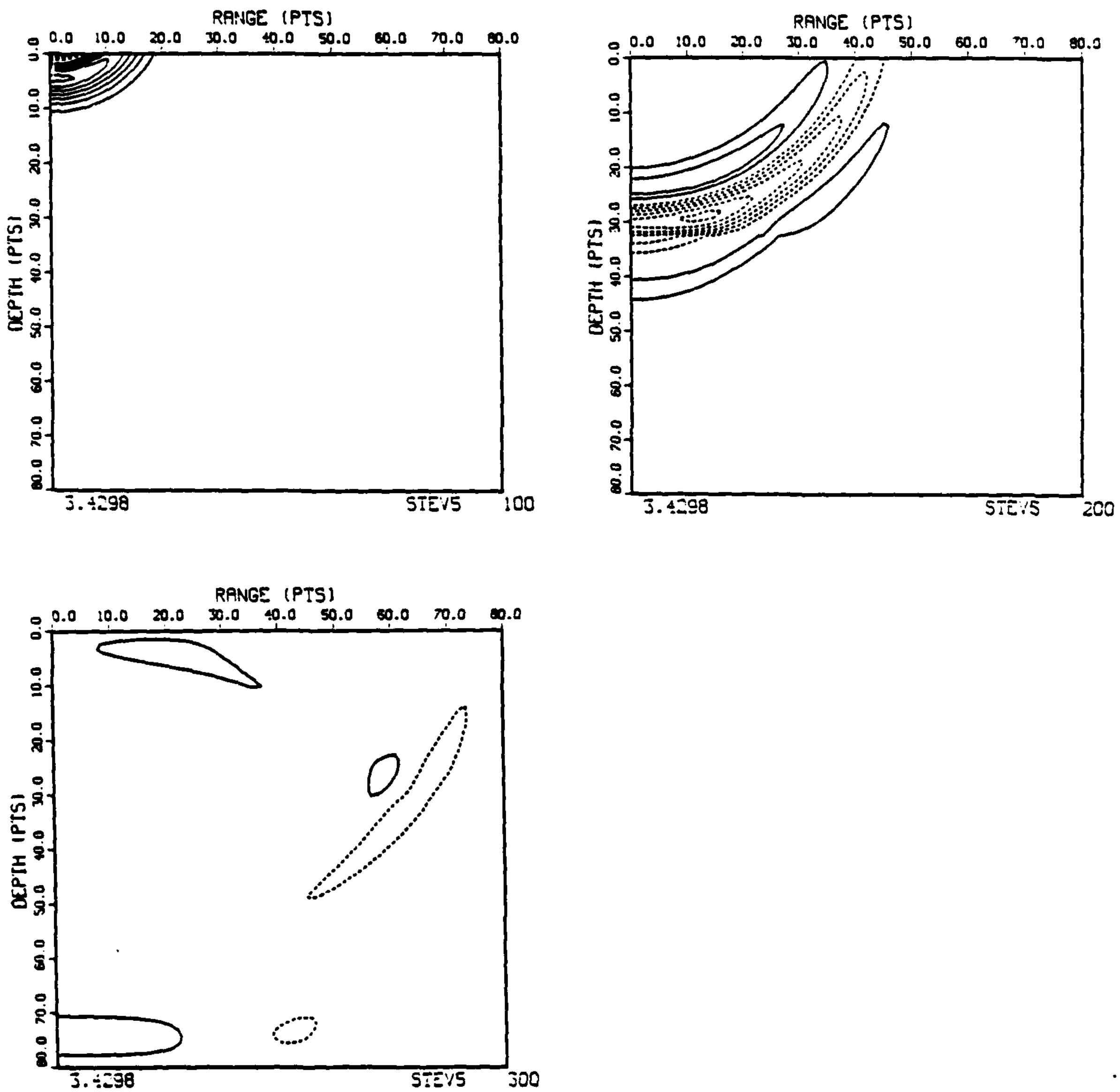


Figure 8

Contour plots of snap-shot files from model STEVA

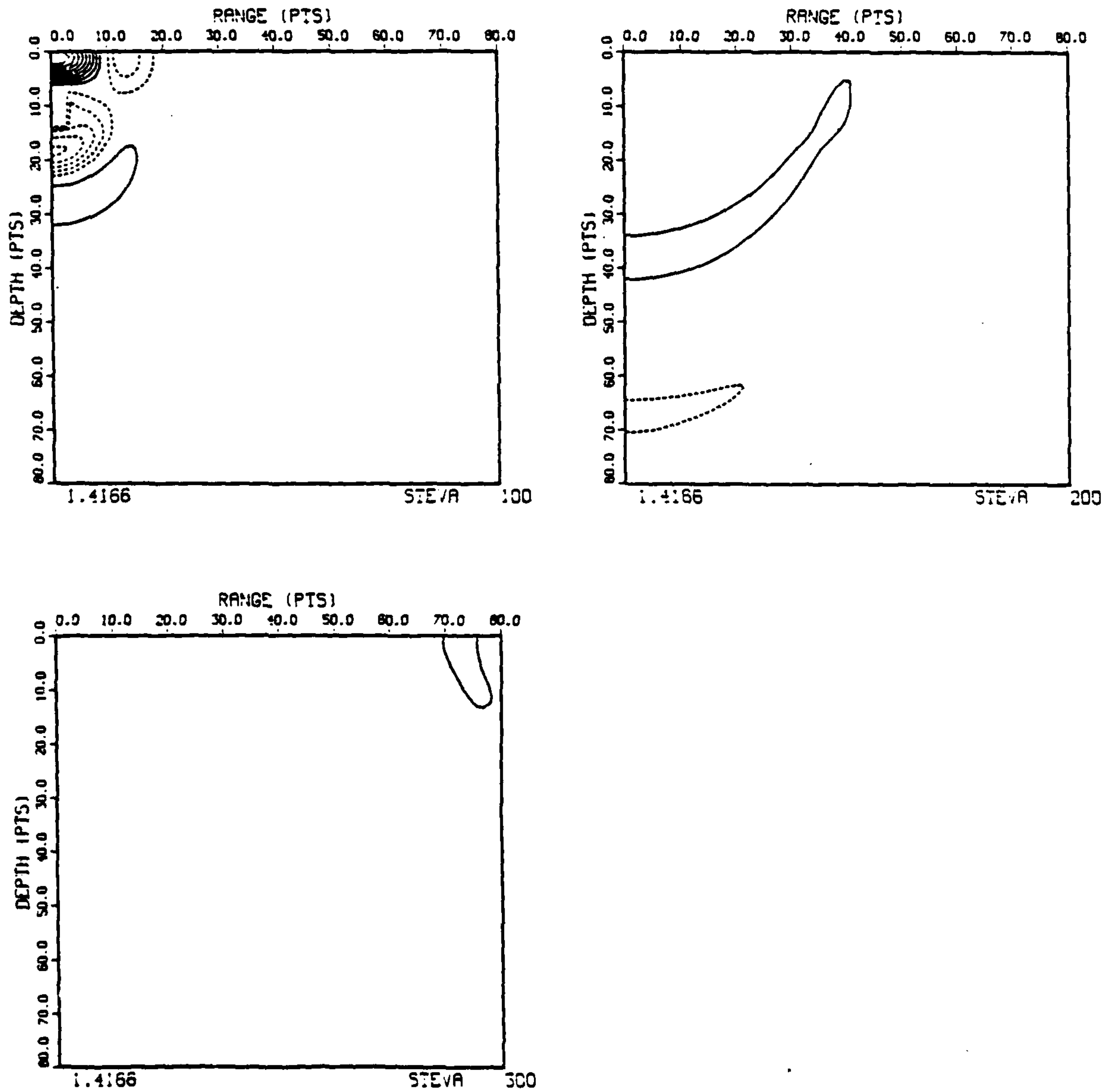


Figure 9

Contour plots of snap-shot files from model STEVB

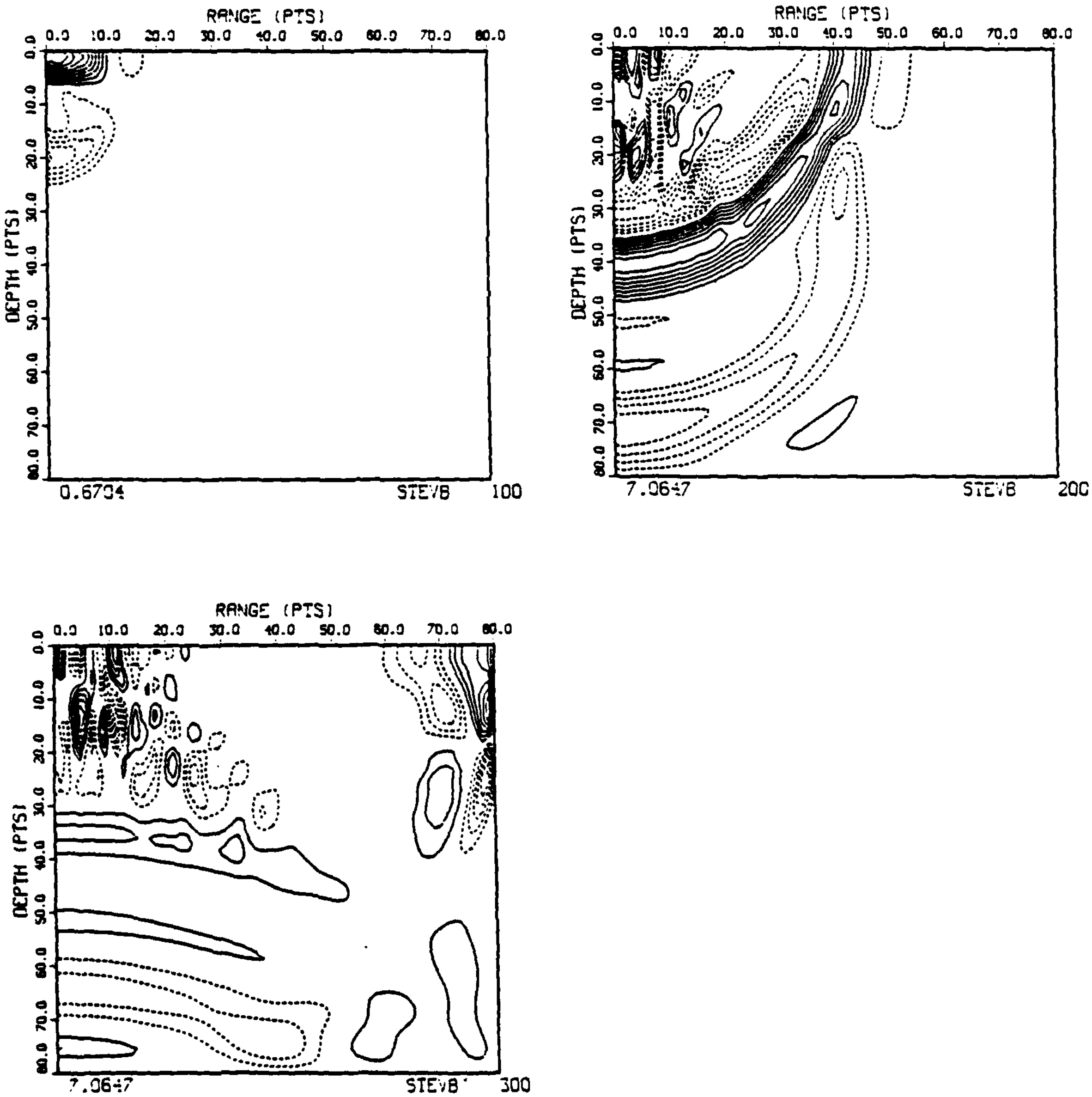


Figure 10

Contour plots of snap-shot files from model STEVC

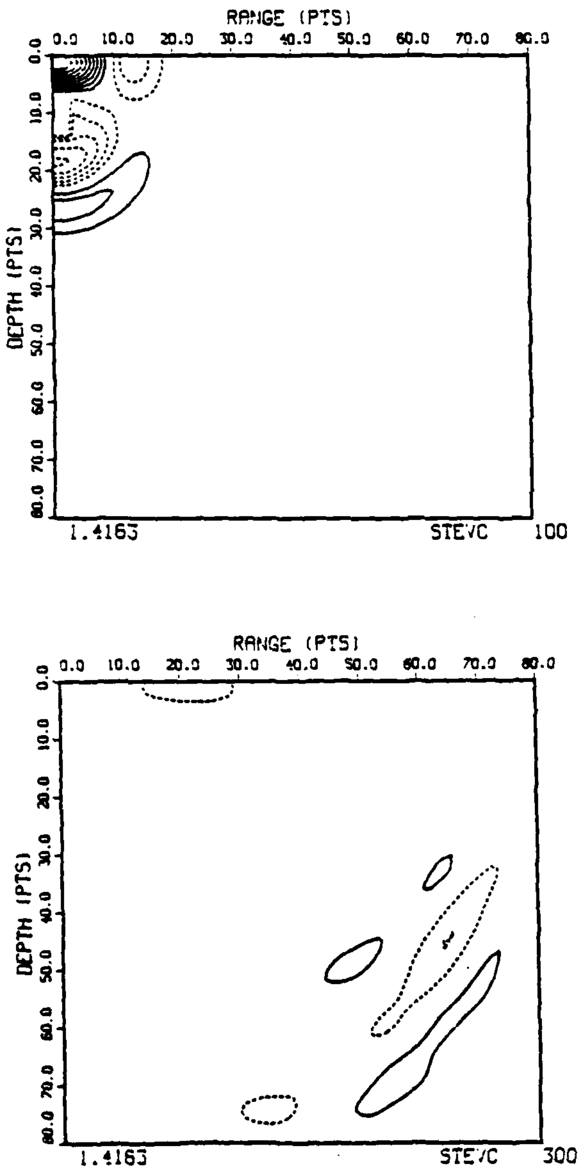


Figure 11

Contour plots of snap-shot files from model STEVD

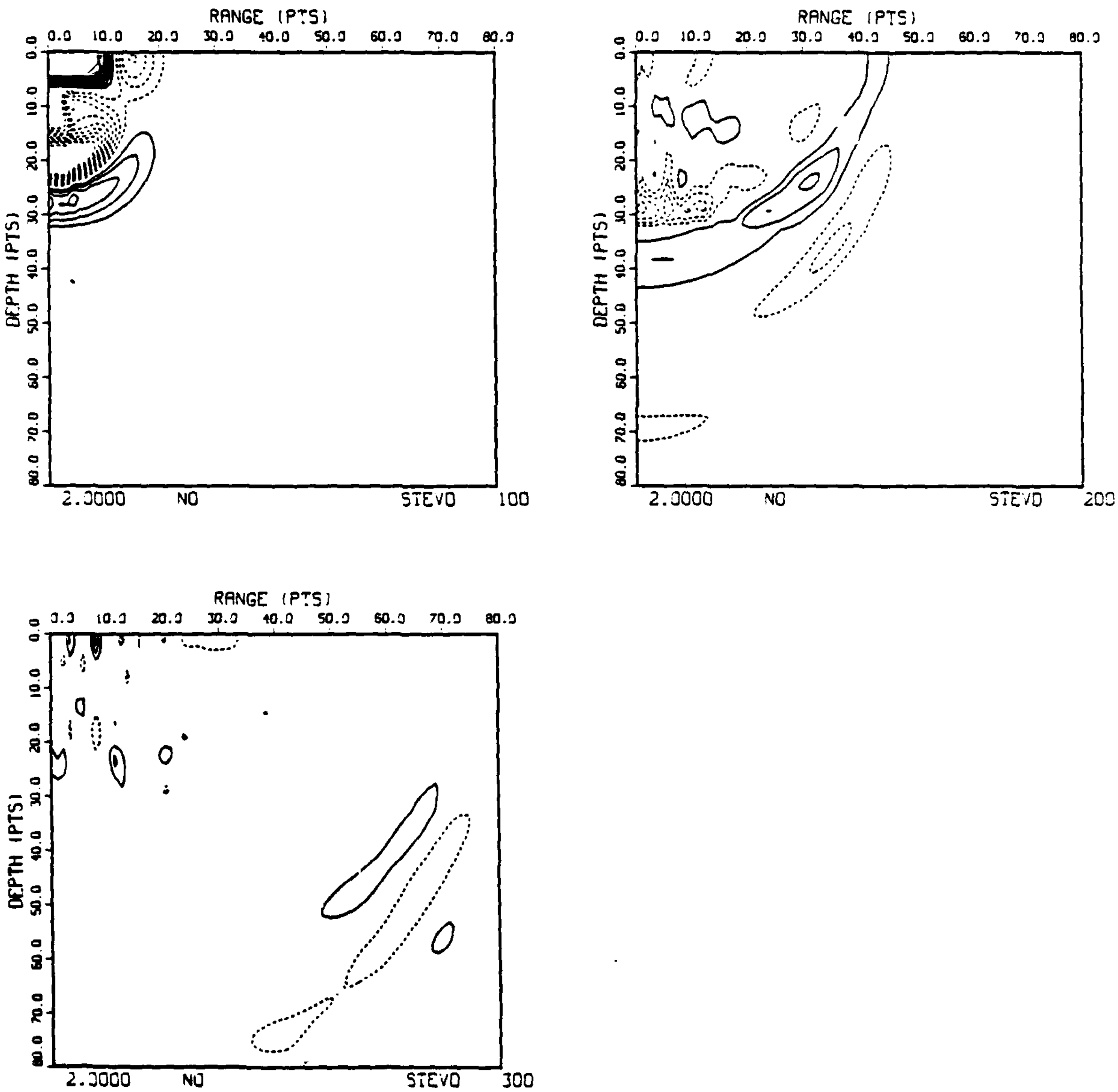
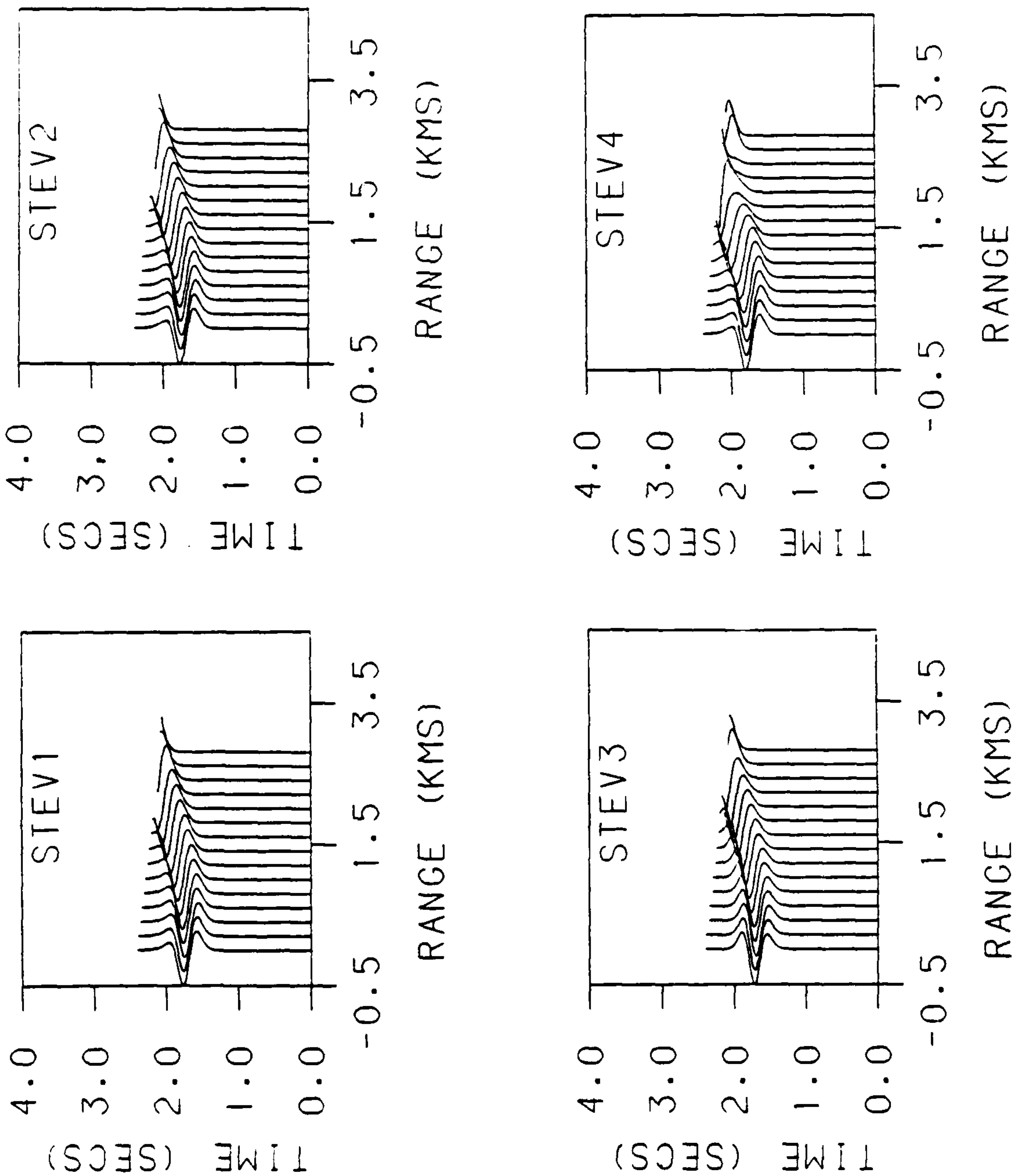


Figure 12

Plots of time series files of STEV1, STEV2, STEV3, and STEV4



X. Notes on Parameter Specifications

The values used in the parameter file cannot be chosen arbitrarily. For stability of the code

$$\Delta t \leq \frac{\min(\Delta r, \Delta z)}{\left[\sqrt{\alpha^2 + \beta^2}\right]_{max.}} \quad , \quad \alpha = \sqrt{\frac{\lambda + 2\mu}{\rho}} \quad , \quad \beta = \sqrt{\frac{\mu}{\rho}} \quad .$$

where α and β are compressional and shear wave speeds. For acceptable grid dispersion one should choose at least ten grid points per wavelength at the upper half power frequency. That is

$$\Delta x \leq \frac{\lambda_{1/2}}{10} = \frac{1}{10} \frac{\alpha_{min}}{f_{+1/2}} \quad .$$

For the Kelly source (the Alterman source is not recommended) the center frequency for displacement is $f_0 = 0.318\sqrt{\xi}$ and the upper half power frequency is $f_{+1/2} = 0.458\sqrt{\xi}$.

Codes without density variations have not yet been written so IDENS is always one. Also the absorbing boundaries only work for rectangular co-ordinates. This program contains only explicit codes so IEXPL is always one. When IFLAT equals one the source is introduced along the top of the grid as if the source were above the box (ND is negative) and NSW and MSW are not used.

When IFLAT equals zero the source is introduced on the edges of an MSW X NSW box at the axis of symmetry centered at depth ND (positive). TSWAVE should be chosen so that the initial disturbance at the grid is small ($\sim 10^{-6}$). Examples of a number of different options are included in the test runs.

XI. References

1. Alterman, Z. S. and Karal, F. C. Jr. 1968 BSSA,58, 367-398.

2. Alterman, Z. S. and Loewenthal, D. 1972 Methods in Computational Physics v.12 Alder, B. et al, Editors

3. Ilan, A., Ungar, A. and Alterman, Z. 1975 GJAS,43, 727-745.

4. Kelly et al. 1976 Geophysics, 41 2 - 27

5. Ilan, A. and Loewenthal, D. 1976 Geophys. Prosp., 24 431, 453

6. Stephen, R. A. 1983 Geophys. J. R. astr. Soc., 72, 39-58.

7. Emerman, S. N. and Stephen, R. A. 1983. Bull. seis. Soc. Am., 73, 661-665.

APPENDIX

Performance Evaluation

The calculation of synthetic seismograms by finite difference methods requires significant amounts of CPU time and computer memory. Throughout the program development we have tried to minimize the resources required, to enable us to run larger and more complex models at reasonable cost. During March, 1981 a study was conducted to determine how the relation between working set size and program size affects CPU time and page faults. Three different sized models (1.2 Mbyte, 1.8 Mbyte, and 2.7 Mbyte) were run with working set sizes varying from 400 to 4000 pages. Table V summarizes the results.

It will be noted that, for each size model, there is a critical working-set size, where the number of page faults is drastically reduced. Nothinq is gained by increasing the working-set size above this critical size. It will also be noted that the difference between the program size and the critical working-set size is the same for each model. This constant, we can assume, is the size of the code which is only executed once, while the critical working-set size includes the arrays used in the computation and the repeated code. Since the largest model in these tests is still a small model, and the 4000 page working-set required was pushing the limit of the VAX, it is obvious that some drastic steps would be required to run even moderately large models economically on the VAX.

A second group of tests was made to determine the effect of using larger arrays than needed for the model. A model which would have required 1.5 Mbytes was run in a program of 1.8 Mbytes and one of 2.7 Mbytes. The results are given in Table VI. Although the working sets used were below the critical size in each case, it can be seen that CPU time was reduced substantially by using the smaller arrays. This is the reason for recompiling and relinking the programs for each model.

Table V

Relation between program size, working-set size, and page faults.

PAGE FAULTS	CPU TIME	PEAK WORKING SET SIZE	DEFAULT WORKING SET SIZE
2997	6.92	400	400
2258	6.54	600	600
2229	6.55	800	800
1271	6.08	1000	1000
1289	6.03	1129	1200
1240	6.07	1130	1400
1238	6.17	1131	1600
1227	7.13	1141	1800
1353	6.99	1141	2000

1.2 Mbyte model (2392 pages)

PAGE FAULTS	CPU TIME	PEAK WORKING SET SIZE	DEFAULT WORKING SET SIZE
33705	75.81	500	500
28458	74.53	1000	1000
25887	72.97	1500	1500
25790	73.06	2000	2000
2443	63.24	2318	2500
2442	63.63	2322	3000
2442	63.63	2326	3500
2442	64.71	2331	4000

1.8 Mbyte model (3595 pages)

PAGE FAULTS	CPU TIME	PEAK WORKING SET SIZE	DEFAULT WORKING SET SIZE
41637	134.81	500	500
41121	135.45	1000	1000
36857	134.73	1500	1500
30381	135.60	2000	2000
30695	133.77	2500	2500
34092	130.09	3000	3000
41383	124.15	3500	3500
4189	102.17	3998	4000

2.7 Mbyte model (5267 pages)

Table VI

PAGE FAULTS	CPU TIME	PEAK WORKING SET SIZE	DEFAULT WORKING SET SIZE
35537	83.91	200	200
30114	84.36	400	400
31200	81.57	600	600
29764	82.33	800	800
28103	84.62	1000	1000
25655	85.16	1200	1200
23946	84.76	1400	1400
22867	83.98	1600	1600
21394	82.61	1800	1800
23176	83.68	2000	2000

1.5 Mbyte model in 2.7 Mbyte program (5267 pages)

PAGE FAULTS	CPU TIME	PEAK WORKING SET SIZE	DEFAULT WORKING SET SIZE
35660	78.57	200	200
34002	78.41	400	400
33617	77.99	600	600
31749	76.35	800	800
26455	74.89	1000	1000
25930	75.20	1200	1200
25590	73.97	1400	1400
25913	74.66	1600	1600
25666	74.19	1800	1800
25767	73.84	2000	2000

1.5 Mbyte model in 1.8 Mbyte program (3595 pages)

As a result of these studies it became clear that more computing power would be necessary to run "real-world" models. The first step was to increase the physical memory of the VAX from 2.5 to 4.0 megabytes. The performance when running small to moderate models was improved, but it was still not reasonable to run large models.

Array Processors were then explored as a means of improving performance. The Floating Point Systems family of array processors was chosen as a possibility and a series of bench-marks was run. Table VII summarizes the results. It is obvious that the VAX/AP performance is degraded by the excessive amount of I/O between the VAX and the AP. In fact, even moderate-sized models are I/O bound. Until the I/O bandwidth is increased, or sufficient AP memory is available, the array processor is not the solution.

The only remaining option seemed to be the use of a super computer. In December 1982, a contract was signed with Control Data Corporation to establish a CYBERNET SERVICES link to a CYBER-205. The initial code conversion was performed by Stu Gould of C.D.C.; he also provided vectorized code for two main loops in the TSTEP routine. Work began in February 1983 in Woods Hole via a 300 BAUD dial-up phone line. After operational details had been ironed out, successful runs of both the scalar and vector codes were made. A comparison of CYBER-205 and VAX timing and cost is shown in Table VIII.

It must be realized that while these figures provide a "ball-park" idea of efficiency, they are not exactly comparable. VAX charges reflect only CPU cost while CYBER charges are total costs. This error is about 10 . The vector code becomes more efficient as the model size increases, leading us to expect more favorable costs for a larger model. Finally, the CYBER-205 runs were done during prime shift, which adversely affects wall-clock time, while the VAX run was non-prime shift on a dedicated machine, optimizing wall-clock time.

Since that time, work has been progressing toward implementing a later version of the programs in the CYBER-205. Preliminary tests indicate that the unvectorized code runs about 60 times faster on the CYBER for large models. We hope for even further savings by completing the vectorization of the TSTEP routines.

Table VII

PERFORMANCE SUMMARY

PARAMETER FILE[1]	VAX/AP WALL TIME[2]	VAX/AP CPU TIME	VAX WALL TIME	VAX CPU TIME	PERFORMANCE IMPROVEMENT[3] WALL	PERFORMANCE IMPROVEMENT[3] CPU	DIRECT I/O COUNT VAX/AP	DIRECT I/O COUNT VAX
BM005	57	2.20	98	96.13	1.72	43.70	162	10
BM006	92	19.14	99	97.38	1.08	5.09	1862	36
BM008	312	136.12	121	117.64	0.39	0.86	11546	27

NOTE: 1. FDAP was executed with parameter file BM005 and BM006. FDAPSEG was executed with parameter file BM008.

2. All times are in seconds.

3. This is defined as VAX time divided by VAX/AP time.

CONFIGURATION STATISTICS

VAX 11/780

VMS V2.01

Default Working Set = 150

1.25 MBYTE Memory

No Other Users

AP-120B

APFORTRAN V2.02

64K Words 167ns. Main Data Memory

4K Words Program Source Memory

Table VIII

	CPU SECONDS	WALL CLOCK SECONDS	PRIME SHIFT COST	NON-PRIME SHIFT COST
VAX 11/780 (2 megabytes)	857.3	995.7	$28.30	$14.15
CYBER-205 (Scalar)	23.2	88.0	$27.81	$14.42
CYBER-205 (vector)	17.6	27.0	$21.87	$11.34

50272-101

REPORT DOCUMENTATION PAGE	1. REPORT NO. WHOI-83-42	2. AD [illegible] 104	3. Recipient's Accession No.

4. Title and Subtitle
FINDIF: A Software Package to Create Synthetic Seismograms by Finite Differences

5. Report Date
November 1983

6.

7. Author(s)
Mary M. Hunt , Lee Gove, and Ralph A. Stephen

8. Performing Organization Rept. No.
WHOI-83-42

9. Performing Organization Name and Address
Woods Hole Oceanographic Institution
Woods Hole, Massachusetts 02543

10. Project/Task/Work Unit No.
NR 083-004

11. Contract(C) or Grant(G) No.
(C) N00014-79-C-0071;
(G)

12. Sponsoring Organization Name and Address
Office of Naval Research
Environmental Sciences Directorate
Arlington, Virginia 22217

13. Type of Report & Period Covered
Technical

14.

15. Supplementary Notes
This report should be cited as: Woods Hole Oceanog. Inst. Tech. Rept. WHOI-83-42.

16. Abstract (Limit: 200 words)

In order to study seismic wave propagation through laterally varying sea floor structures, a software package has been created to generate synthetic seismograms by finite differences. The elastic wave equation can be solved in two dimensions either for point sources in cylindrical coordinates or for line sources in rectangular coordinates. Vertical and radial variations of the elastic parameters are allowed.

The package includes *four programs*. Input to the system consists of a short file containing parameter values to describe the model. The first program is used to initialize the system for the particular model being used. The source arrays and velocity matrices are each computed by a separate program. The final program, which actually carries out the finite difference calculations, includes six subroutines to implement different options based on alternative finite difference formulations. Two different kinds of output files are created by this program: one or more snapshot files, and one time series file, which will usually include more than one series.

17. Document Analysis a. Descriptors
1. Synthetic seismograms
2. Elastic wave equation
3. Finite differences

b. Identifiers/Open-Ended Terms

c. COSATI Field/Group

18. Availability Statement:	19. Security Class (This Report)	21. No. of Pages
Distribution Unlimited	Unclassified	45
	20. Security Class (This Page)	22. Price

(See ANSI-Z39.18) See Instructions on Reverse OPTIONAL FORM 272 (4-77)
(Formerly NTIS-35)
Department of Commerce

MANDATORY DISTRIBUTION LIST

FOR UNCLASSIFIED TECHNICAL REPORTS, REPRINTS, AND FINAL REPORTS PUBLISHED BY OCEANOGRAPHIC CONTRACTORS OF THE OCEAN SCIENCE AND TECHNOLOGY DIVISION OF THE OFFICE OF NAVAL RESEARCH

(Revised October 1983)

1	Deputy Under Secretary of Defense (Research and Advanced Technology) Military Assistant for Environmental Science Room 3D129 Washington, DC 20301
	Office of Naval Research 800 North Quincy Street Arlington, VA 22217
3	Attn: (Code applicable to Program) *
1	Attn: Code 420C
2	Attn: Code 102C
	Commanding Officer Naval Research Laboratory Washington, DC 20375
6	Attn: Library Code 2627
1	Attn: Library Code 2620, Mr. Peter Imhof
12	Defense Technical Information Center Cameron Station Alexandria, VA 22314 Attn: DCA
	Commander Naval Oceanographic Office NSTL Station Bay St. Louis, MS 39522
1	Attn: Code 8100
1	Attn: Code 6000
1	Attn: Code 3300
1	NODC/NOAA Code D781 Wisconsin Avenue, N.W. Washington, DC 20235

* Applicable Codes: 422 (PO); 422CB (Chem/Bio); 422CS (Coastal); 425 (G&G); 425AR (Arctic); 421 (OE); 421SP (Ships); 425OA (Ocean Acoustics); 425UA (Underwater Acoustics)